THREAD GROUP IN JAVA

Java provides a convenient way to group multiple threads in a single object. In such way, we can suspend, resume or interrupt group of threads by a single method call. ThreadGroup creates a group of threads. It offers a convenient way to manage groups of threads as a unit. This is particularly valuable in situation in which you want to suspend and resume a number of related threads.

• The thread group form a tree in which every thread group except the initial thread group has a parent.

• A thread is allowed to access information about its own thread group but not to access information about its thread group's parent thread group or any other thread group.

Constructors of ThreadGroup class

There are only two constructors of ThreadGroup class.

Constructor Description

> ➢ Thread Group (String name) creates a thread group with given name.
> ➢ Thread Group (ThreadGroup parent, String name) creates a thread group with given parent group and name.

The following program is an example for ThreadGroup

public class ThreadGroup_example implements Runnable{

public void run() {

System.out.println(Thread.currentThread().getName());

}

public static void main(String[] args) {

ThreadGroup_example runnable = new ThreadGroup_example();

ThreadGroup tg1 = new ThreadGroup("Parent ThreadGroup");

Thread t1 = new Thread(tg1, runnable,"one");

t1.start();

Thread t2 = new Thread(tg1, runnable,"two");

t2.start();

Thread t3 = new Thread(tg1, runnable,"three");

t3.start();

System.out.println("Thread Group Name: "+tg1.getName());

tg1.list();

}

}

Sample Output:

one

two

three

Thread Group Name: Parent ThreadGroup

java.lang.ThreadGroup[name=Parent ThreadGroup,maxpri=10]

Thread [one,5,Parent ThreadGroup]

Thread [two,5,Parent ThreadGroup]

Thread [three,5,Parent ThreadGroup]

Methods of ThreadGroup class

There are many methods in ThreadGroup class. A list of important methods are given below.

Method Description

- ➢ int activeCount() - returns no. of threads running in current group.
- ➢ int activeGroupCount() - returns a no. of active group in this thread group.
- ➢ void destroy() - destroys this thread group and all its sub groups.
- ➢ String getName() - returns the name of this group.
- ➢ ThreadGroup getParent() - returns the parent of this group.
- ➢ void interrupt() - interrupts all threads of this group.
- ➢ void list() - prints information of this group to standard console.

The following programs explain the threadgroup example.

// Java code illustrating activeCount() method

```
import java.lang.*;

class NewThread extends Thread

{

NewThread(String threadname, ThreadGroup tgob)

{

super(tgob, threadname);

start();

}

public void run()

{

for (int i = 0; i < 1000; i++)

{

try

{

Thread.sleep(10);

}

catch (InterruptedException ex)

{

System.out.println("Exception encounterted");

}

}

}

public class ThreadGroup_example
```

```
{
public static void main(String arg[])
{
// creating the thread group
ThreadGroup gfg = new ThreadGroup("parent thread group");
NewThread t1 = new NewThread("one", gfg);
System.out.println("Starting one");
NewThread t2 = new NewThread("two", gfg);
System.out.println("Starting two");
// checking the number of active thread
System.out.println("number of active thread: "
+ gfg.activeCount());
}
}
```

Output:

Starting one

Starting two

number of active thread: 2